Claims

What is claimed is:

1. A method for storing a data set having an enabled probe identification component and an associated data component, comprising:

   obtaining data from an instrumented program using a probe;

   associating the data with an enabled probe identification; and

   storing the data in the data set, wherein the enabled probe identification is stored in the enabled probe identification component and the data is stored in the associated data set component.

2. The method of claim 1, further comprising:

   defining a tracing function wherein the tracing function comprises an action;

   associating the action with the enable probe identification; and

   associating the probe with the enabled probe identification.

3. The method of claim 2, wherein the tracing function is defined by a consumer.

4. The method of claim 3, wherein the enabled probe identification is defined on a per-consumer basis.

5. The method of claim 1, further comprising:

   associating the enabled probe identification with metadata.

6. The method of claim 5, wherein the metadata defines the layout of the data.

7. The method of claim 5, wherein the metadata includes at least one selected from the group consisting of an action name, a module name, a data size, a data type, and an action function.

8. The method of claim 1, wherein the enabled probe identification is associated with metadata.

9. The method of claim 1, wherein the data set is stored in a kernel-level buffer.

10. A method for processing a data set, comprising:

copying the data set to a user-level buffer, wherein the data set comprises an enabled probe identification and data;

obtaining the enabled probe identification from the data set;

obtaining metadata using the enabled probe identification; and

processing the data set using the data and the metadata.

11. The method of claim 10, wherein the metadata defines the layout of the data.

12. The method of claim 10, wherein the metadata includes at least one selected from the group consisting of an action name, a module name, a data size, a data type, and an action function.

13. A system for storing a data set, wherein the data set comprises an enabled probe identification component and a data component, comprising:

a probe obtaining data from an instrumented program;

a tracing framework associating the probe with an enabled probe identification; and

a buffer storing the data set, wherein the data is stored in the data component and the enabled probe identification is stored in the enabled probe identification component.

14. The system of claim 13, further comprising:

a consumer defining an action, wherein the tracing framework assigns the enabled probe identification to the action.

15. The system of claim 13, further comprising:

an EPID-Metadata table relating the enabled probe identification to metadata.

16. The system of claim 15, wherein the metadata includes at least one selected from the group consisting of an action name, a module name, a data size, a data type, and an action function.

17. The system of claim 14, wherein the enabled probe identification is defined with respect to the consumer.

18. A system for storing a data set, wherein the data set comprises an enabled probe identification component and a data component, comprising:

a probe obtaining data from an instrumented program;

a tracing framework assigning an enabled probe identification to an action and associating the probe with the enabled probe identification; and

a per-consumer buffer storing the data set,

wherein the data is stored in the data component and the enabled probe identification in the enabled probe identification component, and

wherein the enabled probe identification is assigned to the action defined by the consumer associated with the per-consumer buffer.

19. The system of claim 18, further comprising:

an EPID-Metadata table relating the enabled probe identification to metadata.

20. The system of claim 19, wherein the metadata includes at least one selected from the group consisting of an action name, a module name, a data size, a data type, and an action function.